

- [24] S. Narayan, C. Disselkoen, D. Moghimi, S. Cauligi, E. Johnson, Z. Gang, A. Vahldiek-Oberwagner, R. Sahita, H. Shacham, D. Tullsen, *et al.*, “Swivel: Hardening webassembly against spectre,” in *USENIX Security Symposium*, 2021.
- [25] Q. Stiévenart, C. De Roover, and M. Ghafari, “Security risks of porting c programs to webassembly,” in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, SAC ’22, (New York, NY, USA), p. 1713–1722, Association for Computing Machinery, 2022.
- [26] “Global CDN Disruption.” <https://www.fastly.com/blog/summary-of-june-8-outage>, 2021.
- [27] “Stop a wasm compiler bug before it becomes a problem | fastly.” <https://www.fastly.com/blog/defense-in-depth-stopping-a-wasm-compiler-bug-before-it-became-a-problem>, 2021.
- [28] J. Cabrera Arteaga, O. Floros, O. Vera Perez, B. Baudry, and M. Monperrus, “Crow: code diversification for webassembly,” in *MADWeb, NDSS 2021*, 2021.
- [29] H. Okhravi, M. Rabe, T. Mayberry, W. Leonard, T. Hobson, D. Bigelow, and W. Streilein, “Survey of cyber moving targets,” *Massachusetts Inst of Technology Lexington Lincoln Lab, No. MIT/LL-TR-1166*, 2013.
- [30] F. B. Cohen, “Operating system protection through program evolution,” *Computers & Security*, vol. 12, no. 6, pp. 565–584, 1993.
- [31] S. Forrest, A. Somayaji, and D. Ackley, “Building diverse computer systems,” in *Proceedings. The Sixth Workshop on Hot Topics in Operating Systems (Cat. No.97TB100133)*, pp. 67–72, 1997.
- [32] L. Davi, C. Liebchen, A.-R. Sadeghi, K. Z. Snow, and F. Monrose, “Isomeron: Code randomization resilient to (just-in-time) return-oriented programming,” in *NDSS*, 2015.
- [33] E. G. Barrantes, D. H. Ackley, S. Forrest, T. S. Palmer, D. Stefanovic, and D. D. Zovi, “Randomized instruction set emulation to disrupt binary code injection attacks,” in *Proc. CCS*, pp. 281–289, 2003.
- [34] M. Chew and D. Song, “Mitigating buffer overflows by operating system randomization,” Tech. Rep. CS-02-197, Carnegie Mellon University, 2002.
- [35] A. Romano, D. Lehmann, M. Pradel, and W. Wang, “Wobfuscator: Obfuscating javascript malware via opportunistic translation to webassembly,” in *2022 IEEE Symposium on Security and Privacy (SP)* (SP), (Los Alamitos, CA, USA), pp. 1101–1116, IEEE Computer Society, may 2022.

- [36] J. Cabrera Arteaga, “Artificial software diversification for webassembly,” 2022. QC 20220909.
- [37] A. Rossberg, B. L. Titzer, A. Haas, D. L. Schuff, D. Gohman, L. Wagner, A. Zakai, J. F. Bastien, and M. Holman, “Bringing the web up to speed with webassembly,” *Commun. ACM*, vol. 61, p. 107–115, nov 2018.
- [38] A. Hilbig, D. Lehmann, and M. Pradel, “An empirical study of real-world webassembly binaries: Security, languages, use cases,” *Proceedings of the Web Conference 2021*, 2021.
- [39] L. Wagner, M. Mayer, A. Marino, A. Soldani Nezhad, H. Zwaan, and I. Malavolta, “On the energy consumption and performance of webassembly binaries across programming languages and runtimes in iot,” in *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering, EASE '23*, (New York, NY, USA), p. 72–82, Association for Computing Machinery, 2023.
- [40] D. Lehmann, J. Kinder, and M. Pradel, “Everything old is new again: Binary security of webassembly,” in *29th USENIX Security Symposium (USENIX Security 20)*, USENIX Association, Aug. 2020.
- [41] N. Mäkitalo, T. Mikkonen, C. Pautasso, V. Bankowski, P. Daubaris, R. Mikkola, and O. Beletski, “Webassembly modules as lightweight containers for liquid iot applications,” in *International Conference on Web Engineering*, pp. 328–336, Springer, 2021.
- [42] P. K. Gadepalli, S. McBride, G. Peach, L. Cherkasova, and G. Parmer, “Sledge: A serverless-first, light-weight wasm runtime for the edge,” in *Proceedings of the 21st International Middleware Conference*, p. 265–279, 2020.
- [43] R. Gurdeep Singh and C. Scholliers, “Warduino: A dynamic webassembly virtual machine for programming microcontrollers,” in *Proceedings of the 16th ACM SIGPLAN International Conference on Managed Programming Languages and Runtimes, MPLR 2019*, (New York, NY, USA), pp. 27–36, ACM, 2019.
- [44] I. Bastys, M. Algehed, A. Sjösten, and A. Sabelfeld, “Secwasm: Information flow control for webassembly,” in *Static Analysis* (G. Singh and C. Urban, eds.), (Cham), pp. 74–103, Springer Nature Switzerland, 2022.
- [45] T. Brito, P. Lopes, N. Santos, and J. F. Santos, “Wasmati: An efficient static vulnerability scanner for webassembly,” *Computers & Security*, vol. 118, p. 102745, 2022.